

Remarks

Applicants respectfully request reconsideration of the present application in view of the foregoing amendments and the following remarks. Claims 1-15, 17-23, 26-32, 34-37, 39-41, and 43 are pending in the application. Claims 1-15, 17-23, 26-32, 34-37, 39-41, and 43 are rejected. No claims have been allowed. Claims 1, 15, 31, 34, 39, and 43 are independent. Claims 1, 31, 34, and 43 have been amended.

Information Disclosure Statement

The Examiner notes alleged deficiencies in the IDS filed on September 16, 2008, and returned multiple previously filed IDSs with several items unconsidered on November 10, 2008. Applicants note, however, that except for the Arnold, Gupta, and Taft references submitted for the Examiner's consideration in an IDS herewith, all of the items in the IDSs had been previously considered by the Examiner, for example on November 26th and 27th, 2006, and October 29th, 2008.

Regarding the Arnold, Gupta, and Taft references, the Examiner has apparently refused to consider them because "the cited document[s are] not referred to by date or place of publication." (Office Action, page 2.) Applicants note that the " place of publication refers to the name of the journal, magazine, or other publication in which the information being submitted was published." (MPEP § 609.04(a)(I), seventh paragraph.) In the IDS submitted herewith, Applicants are including additional citation detail. Therefore, the IDS submitted herewith should fully comply with the requirements of MPEP § 609 and Applicants respectfully request consideration of the references.

Cited Art

The Action cites Koizumi et al., U.S. Patent Publication No. 2002/0026633 (hereinafter "Koizumi") and Radigan, U.S. Patent Publication No. 2004/0098710 (hereinafter "Radigan").

Claim Rejections under 35 U.S.C. § 102

The Action rejects claims 1, 4-13, 15, 17-23, 26, 27, 30-32, 34-37, 39-41, and 43 under 35 USC 102(a) as being allegedly anticipated by Koizumi. Applicants traverse and respectfully

submit the claims are allowable over the cited art. Accordingly, Applicants request that all rejections be withdrawn.

Claims 1 and 4-13 are Allowable Over Koizumi

Amended claim 1 recites one or more computer-readable media with computer-executable instructions for implementing a software development architecture comprising:

- a software development scenario-independent intermediate representation format;

- one or more exception handling models operable to support a plurality of programming language specific exception handling models for a plurality of different source languages;

- a type system operable to represent the type representations of the plurality of different source languages; and

- a code generator operable to generate code targeted for a plurality of execution architectures;

wherein the code generator constructs one or more software development components of software development tools using the software development scenario-independent intermediate representation format, the one or more exception handling models operable to support the plurality of programming language specific exception handling models for the plurality of different source languages, and the type system operable to represent the plurality of different source languages.

Koizumi does not teach or suggest the above recited language of amended claim 1.

Koizumi does not teach or suggest one or more exception handling models operable to support a plurality of programming language specific exception handling models for a plurality of different source languages. The Examiner alleges that Koizumi discloses "one or more exception handling models operable to support a plurality of programming language specific exception handling models for a plurality of different source languages (e.g. see at least FIG. 2, 3102, 3104, and related text.)" Applicants respectfully disagree. The cited portion of Koizumi describes:

FIG. 2 is a diagram showing an exemplary structure of an abstract register machine compiler (generally denoted by 3100) according to the embodiment of the present invention. Referring to the figure, a source program 3102 undergoes at first a syntax analysis and a semantic analysis by a syntax/semantic analysis section 3104 of the compiler 3100 to be translated into an intermediate language program 3106, while information of a variety of symbols used in the source program 3102 is entered in a symbol table 3108.

(Koizumi, para. 0132.)

To the extent that syntax analysis or semantic analysis is performed and information of a variety of symbols used in the source program is entered into a table, this still does not teach, suggest, or even mention "exception handling models" as recited in amended claim 1. In fact, at no point does Koizumi teach, suggest, or even mention "exception handling models" much less "one or more exception handling models operable to support a plurality of programming language specific exception handling models for a plurality of different source languages" as recited in amended claim 1.

Koizumi's description of a "machine language program" does not teach or suggest a code generator that constructs one or more software development components of software development tools using the software development scenario-independent intermediate representation format, the one or more exception handling models . . . for the plurality of different source languages, and the type system operable to represent the plurality of different source languages. The Examiner alleges that Koizumi discloses "wherein the code generator constructs one or more software development components of software development tools (e.g. FIG. 1, 1011, 1016, and related text.)" (Office Action, page 4.) Applicants respectfully disagree. Koizumi describes, "FIG. 1 is a diagram showing an exemplary structure of a translation system for translating a source program into a machine language program, according to an embodiment of the present invention." (Koizumi, para. 0125.) Elements 1011 and 1016 of FIG. 1 disclose a "Machine Language Program." (FIG. 1.) Applicants note that mere description of "machine language" does not disclose the "one or more software development components of software development tools" as recited in amended claim 1. Applicants make no claim to a software development tool *per se*, but rather a code generator which constructs "one or more software development components of software development tools using the software development scenario-independent intermediate representation format, the one or more exception handling models operable to support the plurality of programming language specific exception handling models for the plurality of different source languages, and the type system operable to represent the plurality of different source languages."

Because Koizumi does not teach or suggest each and every element of amended claim 1, claim 1 is allowable over Koizumi. Dependent claims 4-13 are allowable over Koizumi at least because they depend from allowable claim 1. Applicants respectfully request withdrawal of the § 102(a) rejections and allowance of claims 1 and 4-13.

Claims 15, 17-23, 26, 27, and 30 are Allowable Over Koizumi

Independent claim 15 recites a method of creating a target software development tool, the method comprising:

- receiving at least one computer-readable specification specifying functionality specific to one or more software development scenarios;
- creating at least one software development component for the software development tool from the at least one specification;
- integrating the at least one software development component for the software development tool into a software development scenario-independent framework; and
- compiling the at least one software development component and framework to create the target software development tool;

wherein the computer-readable specification comprises functionality for processing an intermediate representation format capable of representing a plurality of different programming languages; and

wherein the intermediate representation format comprises one or more exception handling models capable of supporting a plurality of programming language-specific exception handling models for the plurality of different programming languages.

Koizumi does not teach or suggest the above recited language of independent claim 15.

As described above, to the extent that Koizumi describes syntax analysis or semantic analysis being performed and information of a variety of symbols used in the source program being entered into a table, this still does not teach or suggest "exception handling models" as recited in independent claim 15. In fact, at no point does Koizumi teach, suggest, or even mention "exception handling models" much less "one or more exception handling models capable of supporting a plurality of programming language-specific exception handling models for the plurality of different programming languages" as recited in independent claim 15.

Additionally, Applicants note that machine language, as used in Koizumi in FIG. 1 and specifically elements 1011 and 1016, would not teach or suggest "creating at least one software development component for the software development tool from the at least one specification" as recited in independent claim 15.

Finally, Koizumi does not teach or suggest "compiling the at least one software development component and framework to create the target software development tool" as recited in independent claim 15. The Examiner points to element 1001 of FIG. 1 on page 7 of

the Office Action to allege that Koizumi describes the element of independent claim 15.

Applicants respectfully disagree. Koizumi describes, for example, "When a source program 1004 is inputted to the compiler 1001, the compiler 1001 performs a syntax analysis, a semantic analysis, an optimization processing (described hereinafter) and others on the source program 1004, at the representation level of the source program, to thereby output an ArmCode (Abstract register machine code) program 1005 which is an abstract object program." (Koizumi, para. 0127.) Thus, Applicants understand Koizumi to simply describe a compiler that receives a source program and outputs ArmCode. However, Applicants are not claiming compiling *per se*; rather, independent claim 15 recites "compiling the at least one software development component and framework to create the target software development tool." Koizumi does not teach or suggest this element of independent claim 15.

Because Koizumi does not teach or suggest each and every element of independent claim 15, claim 15 is allowable over Koizumi. Dependent claims 17-23, 26, 27, and 30 are allowable over Koizumi at least because they depend from allowable independent claim 15. Applicants respectfully request withdrawal of the § 102(a) rejections and allowance of claims 15 and 17-23, 26, 27, and 30.

Claims 31 and 32 are Allowable Over Koizumi

Amended independent claim 31 recites a method of creating a target software development tool from a common framework, the method comprising:

- configuring the common framework based on one or more characteristics of the target software development tool;

- integrating software development components comprising one or more characteristics of the target software development tool into the common framework; and

- creating the target software development tool from the integrated common framework;

- wherein the one or more characteristics comprises an input language chosen from a plurality of different programming languages supported by the common framework for the target software development tool; and

- wherein the common framework comprises exception handling models capable of supporting a plurality of programming language-specific exception handling models for the plurality of different programming languages.

Koizumi does not teach or suggest the above recited language of amended independent

claim 31.

As described above, to the extent that Koizumi describes syntax analysis or semantic analysis being performed and information of a variety of symbols used in the source program being entered into a table, this still does not teach or suggest, nor does Koizumi even mention "exception handling models" as recited in amended independent claim 31. In fact, at no point does Koizumi teach, suggest, or even mention "exception handling models" much less a common framework comprising "exception handling models capable of supporting a plurality of programming language-specific exception handling models for the plurality of different programming languages" as recited in amended independent claim 31.

Additionally, Applicants note that machine language, as used in Koizumi in FIG. 1 and specifically elements 1011 and 1016, would not teach or suggest "integrating software development components comprising one or more characteristics of the target software development tool into the common framework; and creating the target software development tool from the integrated common framework" as recited in amended independent claim 31.

Because Koizumi does not teach or suggest each and every element of independent claim 31, claim 31 is allowable over Koizumi. Dependent claim 32 is allowable over Koizumi at least because it depends from allowable independent claim 31. Applicants respectfully request withdrawal of the § 102(a) rejections and allowance of claims 31 and 32.

Claims 34-37 are Allowable Over Koizumi

Amended independent claim 34 recites a method of producing inter-compatible software development tools, the method comprising:

- creating a first software development tool by integrating first software development components into a software development architecture that is operable to support a plurality of different programming languages; and

- creating a second software development tool based on the first software development tool, wherein the second software development tool dynamically links to a binary version of the software development architecture;

- wherein the software development architecture comprises functionality for exception handling models operable to support programming-language specific exception handling models for the plurality of different programming languages, and the software development architecture is used by both the first and second software development tools.

Koizumi does not teach or suggest the above recited language of amended independent

claim 34.

As described above, to the extent that Koizumi describes syntax analysis or semantic analysis being performed and information of a variety of symbols used in the source program being entered into a table, this still does not teach or suggest "exception handling models" as recited in amended independent claim 34. In fact, at no point does Koizumi teach, suggest, or even mention "exception handling models" much less a software development architecture comprising "functionality for exception handling models operable to support programming-language specific exception handling models for the plurality of different programming languages" as recited in amended independent claim 34.

Additionally, Applicants note that machine language, as used in Koizumi in FIG. 1 and specifically elements 1011 and 1016, would not teach or suggest "creating a first software development tool by integrating first software development components into a software development architecture" as recited in amended independent claim 34.

Because Koizumi does not teach or suggest each and every element of amended independent claim 34, claim 34 is allowable over Koizumi. Dependent claims 35-37 are allowable over Koizumi at least because they depend from allowable independent claim 34. Applicants respectfully request withdrawal of the § 102(a) rejections and allowance of claims 34-37.

Claims 39-41 are Allowable Over Koizumi

Independent claim 39 recites a method of modifying a software development tool, the software development tool having been created using a software development architecture that is operable for a plurality of different programming languages and comprising one or more software development components, the method comprising:

- dynamically linking a software development component not present in the software development architecture to a binary version of the software development architecture that is operable for the plurality of different programming languages; and

- creating a modified software development tool from the dynamically linked binary version and the software development component;

- wherein the binary version of the software development architecture comprises functionality for exception handling models operable to support a plurality of programming language specific exception handling models for the plurality of different programming languages used by the modified software

development tool.

Koizumi does not teach or suggest each and every element of independent claim 39.

As described above, to the extent that Koizumi describes syntax analysis or semantic analysis being performed and information of a variety of symbols used in the source program being entered into a table, this still does not teach or suggest "exception handling models" as recited in independent claim 39. In fact, at no point does Koizumi teach, suggest, or even mention "exception handling models" much less a software development architecture comprising "functionality for exception handling models operable to support a plurality of programming language specific exception handling models for the plurality of different programming languages" as recited in independent claim 39.

Additionally, Applicants note that machine language, as used in Koizumi in FIG. 1 and specifically elements 1011 and 1016, would not teach or suggest "creating a modified software development tool from the dynamically linked binary version and the software development component" as recited in independent claim 39.

Because Koizumi does not teach or suggest each and every element of independent claim 39, claim 39 is allowable over Koizumi. Dependent claims 40 and 41 are allowable over Koizumi at least because they depend from allowable independent claim 39. Applicants respectfully request withdrawal of the § 102(a) rejections and allowance of claims 39-41.

Claim 43 is Allowable Over Koizumi

Amended independent claim 43 recites a method of creating a software development tool, the method comprising:

- receiving at least one computer-executable file comprising:

- an intermediate representation capable of representing a plurality of different programming languages and computer executable images;

- one or more exception handling models capable of supporting a plurality of programming language specific exception handling models for the plurality of different programming languages;

- a type system capable of representing the type representations of a plurality of source languages; and

- a code generator capable of generating code targeted for a plurality of execution architectures;

- linking a software development component to the at least one computer-executable file using least one class extension declarations; and

creating the software development tool via the linked software development component and computer-executable file.

Koizumi does not teach or suggest each and every element of independent claim 43.

As described above, to the extent that Koizumi describes syntax analysis or semantic analysis being performed and information of a variety of symbols used in the source program being entered into a table, this still does not teach or suggest "exception handling models" as recited in independent claim 43. In fact, at no point does Koizumi teach, suggest, or even mention "exception handling models" much less a software development architecture comprising "one or more exception handling models capable of supporting a plurality of programming language specific exception handling models for the plurality of different programming languages" as recited in independent claim 43.

Additionally, Applicants note that machine language, as commonly understood by one of ordinary skill in the art and used in Koizumi in FIG. 1 and specifically elements 1011 and 1016, would not teach or suggest "creating the software development tool via the linked software development component and computer-executable file" as recited in independent claim 43.

Because Koizumi does not teach or suggest each and every element of independent claim 43, claim 43 is allowable over Koizumi. Applicants respectfully request withdrawal of the § 102(a) rejection and allowance of claim 43.

Claim Rejections under 35 U.S.C. § 103(a)

The Action rejects claims 2, 3, 14, 28, and 29 under 35 U.S.C § 103(a) as being allegedly unpatentable over Koizumi in view of Radigan. Applicants traverse.

As described above, independent claims 1 and 15, from which dependent claims 2, 3, 15, 28, and 29 depend, are allowable over Koizumi. Radigan does not cure the above cited deficiencies of Koizumi with respect to independent claims 1 and 15. Regardless of whether Radigan describes JIT compilers, as the Examiner alleges, Applicants do not understand Radigan to teach or suggest "exception handling models," much less "exception handling models operable to support a plurality of programming language specific exception handling models for a plurality of different source languages" as described for example in claim 1 and similarly in independent claim 15. Neither does Radigan teach or suggest "software development components" or "software development tools," much less a code generator which constructs "one

or more software development components of software development tools" as described in claim 1 or "creating at least one software development component for the software development tool from the at least one specification" as described in independent claim 15.

Because Koizumi and Radigan, whether considered separately or in combination with each other, do not teach or suggest each and every element of independent claims 1 and 15, claims 1 and 15, and their respective dependent claims 2, 3, 14, 28, and 29, are allowable over Koizumi and Radigan. Applicants respectfully request withdrawal of the § 103(a) rejections and allowance of claims 2, 3, 14, 28, and 29.

Interview Request

If the claims are not found by the Examiner to be allowable, the Examiner is requested to call the undersigned attorney to set up an interview to discuss this application.

Conclusion

The claims should be allowable. Such action is respectfully requested.

Respectfully submitted,

KLARQUIST SPARKMAN, LLP

One World Trade Center, Suite 1600
121 S.W. Salmon Street
Portland, Oregon 97204
Telephone: (503) 595-5300
Facsimile: (503) 595-5301

By /Cory A. Jones/
Cory A. Jones
Registration No. 55,307